REMARKS/ARGUMENTS

Claims have been amended to further clarify that a Bytecode instruction can be executed by a virtual machine to determine a string representation for an object, thereby allowing the string representation to be determined without invoking a method call. As noted in the specification, an inventive Java Bytecode instruction suitable for execution by a virtual machine can be executed to represent Java objects as strings. This allows Java objects to be represented without involving the Java "to_string" method which is conventionally used (Summary of the invention, paragraph 12). It is respectfully submitted that the cited art does NOT teach or suggest a Virtual Machine Bytecode instruction determines a string representation for an object.

In the Final Office Action, the Examiner has rejected claims 4, 5, 7, 8, 11-15 and 17-22 under 35 U.S.C. § 103(a) as being unpatentable over *Peter van der Linden* in view of US Patent No. 6,654,778 (*Blandy et al.*) and further in view of US Patent No. 6,026,485 (*O'Conner et al.*).

The Examiner has asserted that *Linden* discloses retrieving a string representation associated with a Java object, thereby allowing said string representation to be determined. It is noted that string representation of an object can be determined by calling a "to_string" method. Moreover, the Applicant respectfully submits that *Linden* neither discloses nor suggests using a Java Bytecode instruction for retrieving a string representation associated with a Java object, thereby allowing said string representation to be determined without invoking a Java method. In other words, *Linden* does NOT teach or suggest a Bytecode instruction for determining string representation of an object.

In the Final Office Action, The Examiner has also stated that "*Blandy* discloses a Java bytecode instruction suitable for execution by a Java virtual

machine in a Java computing environment that operates to have the functions performed by a Java method without invoking said Java method." Applicant respectfully submits that *Blandy* discloses that an "interpreter is directed by the Bytecode to execute native code to perform the function indicated by the Bytecode" (*Blandy*, col. 5, lines 65-67), and further states that "methods having eight or fewer Bytecodes are considered potential trivial methods that may be replaced with native code, which performs the function of the method" (Blandy, col. 6, lines 42-45). In other words, Blandy teaches replaces methods with native code. However, it is respectfully submitted that Blandy neither discloses nor suggests a Bytecode instruction that can retrieve (or determine) a string representation associated with a Java object.

The Examiner has rejected claims 6 and 8-19 under 35 U.S.C. § 103(a) as being unpatentable over *Linden* and *Blandy* as applied to Claims 1,3,4,5 and/or 7, and further in view of *O'Connor et al.* (US Patent No 6,026,485). The Examiner stated in pertinent part that it would have been obvious to retrieve the string representation of an object as taught by *Linden* without invoking a Java method in Java Virtual machine as taught by *Blandy* through the conventional use of a Java Virtual machine as taught by *O'Connor et al.* The motivation for doing so would have been because the Java Virtual machine is a stack-oriented abstract computing machine, where instructions operate on data at the top of an operand stack and it is of conventional practice to use the Java Virtual machine in this way as suggested by *O'Connor et al.* Applicant respectfully notes that as mentioned in above, *Linden* and *Blandy* neither disclose nor suggest using a Java Bytecode instruction for retrieving a string representation associated with a Java object, thereby allowing said string representation to be determined without invoking a Java method. In this regard, it is respectfully submitted that viewing the Java Virtual machine

as an abstract stack-oriented machine does not render Claims 6 or 8-19 obvious over the cited art. Similarly, it is respectfully submitted that *O'Connor et al*'s teaching of a Java Aload instruction (as pointed out by the Examiner) for loading references does not render the claims obvious over the cited art because an Aload instruction does not teach or even remotely suggest retrieving or determining a string representation associated with a Java object. Rather, an Aload operation loads a reference on the stack.

Based on the foregoing, it is submitted that the claims are patentably distinct over the cited art of record. Additional limitations recited in the independent claims or the dependent claims are not further discussed because the limitations discussed above are sufficient to distinguish the claimed invention from the cited art. Accordingly, Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 500388 (Order No. SUN1P843). Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

R. Mahboubian
Registration No. 44,890

P.O. Box 70250
Oakland, CA 94612-0250
(650) 961-8300